# ABSTRACT

Expert systems have traditionally relied on rule-based logic to simulate decision-making, but modern applications demand greater flexibility and contextual awareness. This project presents a conversational expert system for personalized travel recommendation using large language models (LLMs). The system collects user preferences through a multi-step chat interface and employs Gemini AI to generate contextual travel suggestions. A hybrid natural language processing pipeline extracts location entities via regex and maps them to geographic coordinates using the OpenCage Geocoding API. Results are visualized using Leaflet. The system demonstrates how integrating LLMs with geospatial data can enhance adaptive recommendation capabilities in real-world domains. Presenting, Smart Tour – An **AI Powered Indian Tour Recommendation System** that features **Geospatial Mapping**.

# REQUIREMENT SPECIFICATIONS

## *1. Functional Requirements*

- The system shall accept user inputs: place type, budget, season, and source location.
- The system shall provide personalized destination recommendations using the Gemini API.
- The system shall fetch live destination details from the Gemini API in real time.
- The system shall obtain geolocation coordinates using the OpenCage Geocoder API.
- The system shall visualize destinations on an interactive Leaflet.js map with:
- Colored markers categorized by Indian states
- The system shall display structured destination details including:
  Name, highlights, attractions, budget, season, and travel info
- The system shall include fallback destinations when AI data extraction fails.
- The system shall allow users to interact with map markers for additional travel details.

## *2. Non-Functional Requirements*

- The system shall be responsive and mobile-friendly.
- The system shall render the frontend using Next.js and Material UI for performance and visual design.
- API keys shall be securely managed using .env files.
- The system shall maintain real-time responsiveness (within ~2 seconds for recommendations).
- The system shall be scalable for future enhancements or domains.

## *3. UI Requirements*

- Form for user preferences (place type, budget, season, source).
- “Get Recommendations” button to trigger processing.

- Interactive map with:
  - Custom markers colored by Indian states
  - Clickable markers showing destination details
- Organized card layout for destination information.
- Clean and modern interface using Material UI components.

### *4. Data Requirements*

- AI response must contain:
  - Recommended destination names
  - Budget, season, attractions, highlights

- OpenCage API must return:
  - Latitude and longitude for map plotting

- Fallback data shall be used when API fails or returns insufficient information.

## MOTIVATION

The current travel recommendation landscape **lacks intelligent systems** that personalize trip plans based on **individual budgets, preferences, and seasonal timing**. Most available apps provide static suggestions or pre-designed itineraries, ignoring the dynamic needs of modern travellers. This gap motivated us to develop Smart Tour—an AI-powered platform that delivers real-time personalized travel plans, making trip planning efficient and stress-free.

The AI travel assistant market is projected to grow by over **20% annually**, yet a large segment of users (approx. 60%) still rely on outdated or generic tools. The lack of interactive maps, budget-conscious features, and real-time adaptability limits user satisfaction. In a diverse and tourism-rich country like India, with its varied climates, regional cultures, and economic disparities, personalized planning is even more critical. From affordable domestic getaways to seasonal pilgrimage trips, Smart Tour bridges the gap by merging data-driven insights with intuitive design, offering tailored travel solutions for every Indian traveller.

What truly motivated us was our own frustrating experience while planning group tours during college breaks—spending hours comparing blogs, prices, and locations, only to end up with rigid plans that didn't suit everyone's preferences. We realized that smarter travel planning wasn't just a luxury, but a necessity in today's fast-paced, budget-aware environment. Smart Tour was born out of this shared struggle, aiming to make travel planning **exciting, inclusive, and effortless.**

## DESCRIPTION OF USAGE OF THE PROJECT

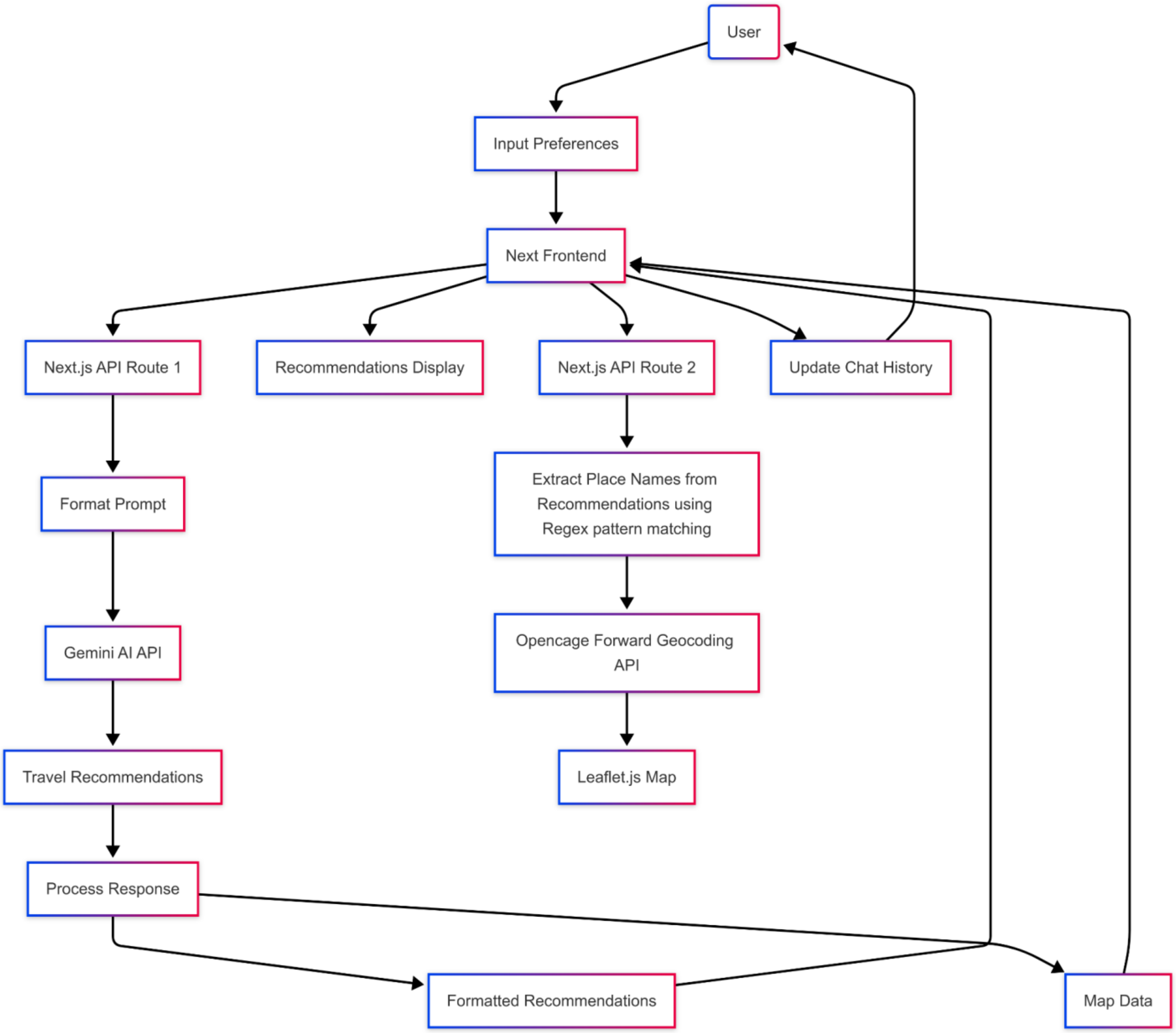
### *How the System Works:*

1. The user accesses the deployed app via a web browser: [https://smart-tour-ecru.vercel.app](https://smart-tour-ecru.vercel.app)

2. A chatbot initiates the session, guiding the user through four main questions:
    - What kind of place do you want to visit? (e.g., Beach, Historical)
    - What is your travel budget?
    - What month or season are you planning to travel?
    - What city will you be traveling from?

3. As the user responds, the system collects and validates the inputs.
4. Once all four inputs are provided, they are packaged and sent to an API endpoint for AI processing.
5. Gemini AI API processes the input and generates personalized travel recommendations.
6. A second API endpoint extracts place names and geocodes them using OpenCage.
7. The extracted places are displayed on an interactive Leaflet.js map.
8. The user is then asked if they want more recommendations, restarting the loop if needed.

*Visual Workflow:*



This system is designed for users who seek a balance between automation and personalization. With a minimal input effort, users can receive high-quality, customized travel suggestions.

# AI IN PERSONALIZED TRAVEL PLANNING

***• Decision Making***

The Gemini API (AI model) analyzes user inputs (interest, budget, season, source) to make intelligent decisions about which Indian destinations are most suitable.

***• Classification***

AI categorizes user preferences into types (e.g., "Beach", "Hill Station", "Adventure", etc.) and matches them with relevant destinations from the knowledge base or API response.

***• Recommendation***

AI generates personalized travel suggestions based on real-time data, ranking and filtering destinations dynamically to suit each user's preferences.

***• Natural Language Understanding***

The Gemini API interprets user queries even when they’re unstructured or conversational, helping to extract structured parameters (like place type, season) for accurate output.

***• Fallback Handling (Robustness)***

If AI fails to parse or respond meaningfully, the system gracefully falls back to default recommendations, ensuring continuous functionality.

***• Dynamic Knowledge Access***

Unlike traditional rule-based expert systems, this AI-based system can fetch and reason with live data, enabling more accurate and updated recommendations.

# DIRECTORY STRUCTURE

```
app/

├──── api/                    # API route handlers

|    ├──── download-pdf/          # Handles PDF generation and download

|    ├──── gemini/                # Integration with Gemini AI for travel suggestions

|    └──── recommendations/        # API to fetch travel destination recommendations

├──── chat/                   # Chat interface page for interacting with the assistant

├──── dashboard/                # Dashboard for visualizing analytics and usage

├──── globals.css               # Global CSS styles applied across the app

├──── layout.tsx                # Root layout defining page structure and theming

├──── leaflet.css               # Leaflet map library styling overrides

└──── page.tsx                  # Main homepage entry point │    └──── page.tsx
```

```
│

├── components/          # Reusable UI components

├── hooks/               # Custom React hooks

├── public/              # Static assets

├── package.json         # Project dependencies

└── tailwind.config.ts   # Tailwind CSS configuration
```

## SOURCE CODE

### *Gemini API Logic:*

```
const response = await fetch(
  `https://generativelanguage.googleapis.com/v1beta/models/${model}:generateContent?key=${apiKey}`,
  {
    method: "POST",
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify({
      contents: [{ parts: [{ text: prompt }] }],
      generationConfig: { temperature: 0.7, topK: 40, topP: 0.95, maxOutputTokens: 4096 }
    })
  }
);
```

### *Extraction of Places from Gemini Response using Custom Regex Logic:*

```
async function extractPlacesFromResponse(response: string) {
  const places = []
  const lines = response.split("\n")

  for (const line of lines) {
    if (/^\d+\./.test(line)) {
      const parenthesesMatch = line.match(/\(([^)]+)\)/)

      if (parenthesesMatch) {
        const stateMatch = line.match(/^\d+\.\s*([^(]+)/)
        const stateName = stateMatch ? stateMatch[1].trim() : "Unknown"
        const placesText = parenthesesMatch[1].trim()

        const placesList = extractIndividualPlaces(placesText)
    for (const placeName of placesList) {
     const coordinates = await getCoordinatesFrom(placeName, stateName)
     if (coordinates) {
      places.push({ name: placeName, state: stateName, lat: coordinates.lat, lng: coordinates.lng })
     }
    }
   } else {
    const stateMatch = line.match(/^\d+\.\s*([^,]+)/)
    if (stateMatch) {
     const stateName = stateMatch[1].trim()
     const coordinates = await getCoordinatesFrom(stateName, "India")
     if (coordinates) {
      places.push({ name: stateName, state: "India", lat: coordinates.lat, lng: coordinates.lng })
     }
    }
   }
  }
 }
 return places
}
```

```
function extractIndividualPlaces(placesText: string): string[] {
 placesText = placesText.replace(/\([^)]*\)/g, "") // Remove parentheses
 placesText = placesText.replace(/\s*\*\s*/g, ", ") // Convert * separators to commas

 const placesList = placesText
  .split(/\s*(?:,|\s+&|\s+and\s+)\s*/i) // Split by commas, "&", or "and"
  .map((place) => place.trim())
  .filter(Boolean)

 return placesList
}
```

## *OpenCage API Logic:*

```
async function getCoordinatesFrom(placeName: string, stateName: string) {
 try {
  const apiKey = process.env.OPENCAGE_API_KEY;
  if (!apiKey) {
   console.error("OpenCage API key not configured");
   return null;
  }

  const searchQuery = `${placeName}, ${stateName}, India`;
  const response = await fetch(

`https://api.opencagedata.com/geocode/v1/json?q=${encodeURIComponent(searchQuery)}&key=${apiKey}&limit=1&countrycode=in`
  );
```

```
    if (!response.ok) throw new Error(`OpenCage API error: ${response.status}`);

    const data = await response.json();
    if (data.results?.length > 0) {
      const { lat, lng } = data.results[0].geometry;
      return { lat, lng };
    }

    console.warn(`No coordinates found for ${placeName}`);
    return null;
  } catch (error) {
    console.error(`Error for ${placeName}:`, error);
    return null;
  }
}
```

## ***Fine Tuning of Gemini Model:***

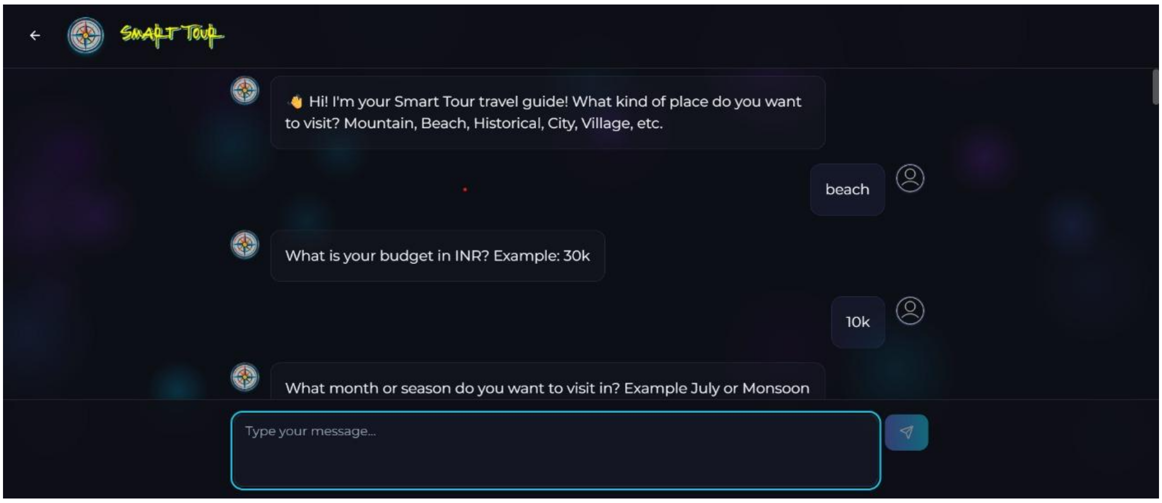
```
const prompt = `
You are a travel advisor specializing in Indian tourism.
Suggest 5 specific destinations in India for ${placeType} places with a budget of ${budget} INR to visit in ${season} from ${source}.

Format your response as follows:

Start with a brief introduction paragraph.

Then list 5 destinations in this format:
1. State Name (Specific places/cities/towns/villages inside parentheses)
• Special: Brief description of what makes this destination special.
• Attractions: Key attractions to visit.
• Budget Considerations: Information about costs.
• Best Season: When to visit and current conditions.
• Ideal Days: Recommended length of stay.
• Travel from ${source}: How to get there.

2. Next Destination(Specific places)
...and so on.

After listing all 5 destinations, include:
• Budget Breakdown (Approximate for 7 Days): with categories like Transportation, Accommodation, Food, etc.
• Travel Tips for ${season}: practical advice for travelers.

IMPORTANT: Make sure to use proper bullet points (•) and avoid using ** or <strong> tags for formatting.
`
```

### *Leaflet.js Mapping of Coordinates:*

```
useEffect(() => {
  if (!mapRef.current) return

  const map = L.map(mapRef.current).setView([20.5937, 78.9629], 5)

  L.tileLayer("https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png", {
    attribution: '&copy; OpenStreetMap contributors',
    maxZoom: 19,
  }).addTo(map)

  const bounds = L.latLngBounds([])

  places.forEach((place) => {
    const marker = L.marker([place.lat, place.lng]).addTo(map)
    marker.bindPopup(`<b>${place.name}</b>${place.state ? `<br>${place.state}` : ""}`)
    bounds.extend([place.lat, place.lng])
  })

  if (places.length > 1 && bounds.isValid()) {
    map.fitBounds(bounds, { padding: [50, 50] })
  } else if (places.length === 1) {
    map.setView([places[0].lat, places[0].lng], 8)
  }

  return () => map.remove()
}, [places])
```

## SAMPLE OUTPUT

### *Input Prompt:*



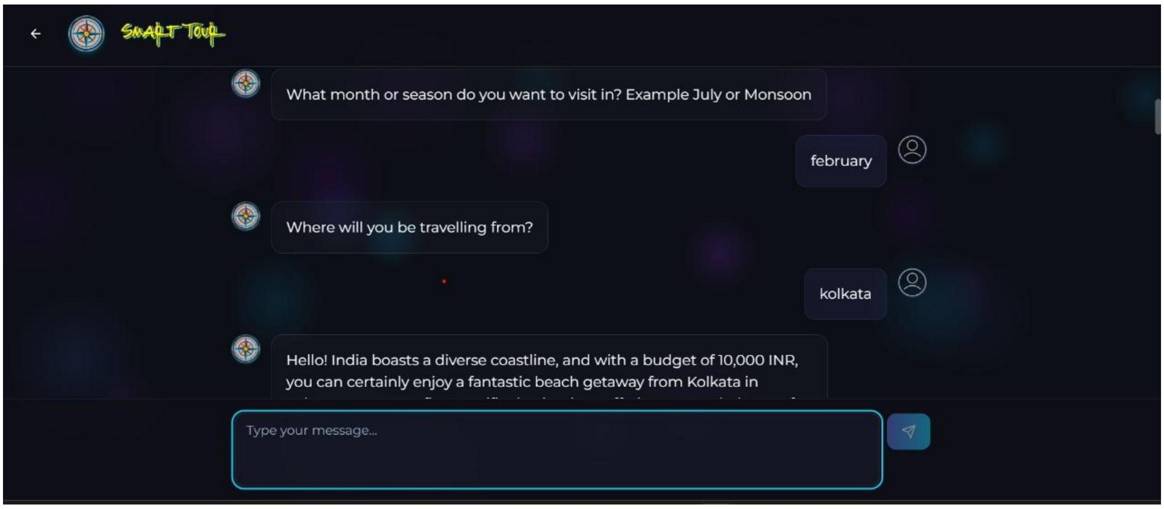
👋 Hi! I'm your Smart Tour travel guide! What kind of place do you want to visit? Mountain, Beach, Historical, City, Village, etc.

beach

What is your budget in INR? Example: 30k

10k

What month or season do you want to visit in? Example July or Monsoon



What month or season do you want to visit in? Example July or Monsoon

february

Where will you be travelling from?

kolkata

Hello! India boasts a diverse coastline, and with a budget of 10,000 INR, you can certainly enjoy a fantastic beach getaway from Kolkata in

### *The Recommendation:*

***“Hello! India boasts a stunning coastline with diverse beaches, perfect for a budget-friendly getaway. February offers pleasant weather across much of the country, making it an ideal time for beach tourism. Here are 5 destinations accessible from Kolkata within a 10k INR budget:***

*1. Odisha (Puri, Konark)*
- ***Special:*** *Known for spiritual significance and beautiful beaches. Puri is home to the Jagannath Temple.*
- ***Attractions:*** *Jagannath Temple (Puri), Puri Beach, Konark Sun Temple, Chandrabhaga Beach.*
- ***Budget Considerations:*** *Odisha is relatively inexpensive. Accommodation and food are affordable, especially if you opt for local guesthouses and street food.*
- ***Best Season:*** *October to March. February offers pleasant temperatures and low humidity.*
- ***Ideal Days:*** *4-5 days.*
- ***Travel from Kolkata:*** *Direct trains are available to Puri from Kolkata (approx. 8-9 hours). Buses are also an option, though less comfortable.*

*2. West Bengal (Digha, Mandarmani)*
- ***Special:*** *Easily accessible from Kolkata, these beaches offer a quick and relaxing escape.*
- ***Attractions:*** *Digha Beach, New Digha Beach, Mandarmani Beach (known for its long motorable stretch), Shankarpur.*
- ***Budget Considerations:*** *Being close to Kolkata, transportation is cheap. Plenty of budget-friendly hotels and food stalls are available.*
- ***Best Season:*** *October to March. February is ideal with comfortable temperatures.*
- ***Ideal Days:*** *2-3 days.*
- ***Travel from Kolkata:*** *Buses and trains operate daily from Kolkata to Digha (approx. 4-5 hours).*

*3. Andhra Pradesh (Visakhapatnam, Araku Valley)*
- ***Special:*** *Offers a blend of beaches and hills, with cultural attractions.*
- ***Attractions:*** *RK Beach, Kailasagiri Hill Park, Submarine Museum, Araku Valley (hill station near Visakhapatnam).*
- ***Budget Considerations:*** *Andhra Pradesh provides value for money. Visakhapatnam offers accommodation options for all budgets.*
- ***Best Season:*** *October to March. February has lovely weather.*
- ***Ideal Days:*** *5-6 days.*
- ***Travel from Kolkata:*** *Direct trains are available to Visakhapatnam from Kolkata (approx. 12-14 hours).*

*4. Tamil Nadu (Mahabalipuram)*
- ***Special:*** *Rich in history and architecture, with beautiful rock-cut temples along the beach.*
- ***Attractions:*** *Shore Temple, Five Rathas, Arjuna's Penance, Mahabalipuram Beach.*
- ***Budget Considerations:*** *Accommodation and food can be found to fit various budgets.*

- ***Best Season:*** *November to February. February is excellent with mild weather.*
- ***Ideal Days:*** *3-4 days.*
- ***Travel from Kolkata:*** *Take a train to Chennai, then a bus or taxi to Mahabalipuram (approx. 1-2 hours). Total train travel around 24-26 Hours.*

***5. Pondicherry (Pondicherry)***

- ***Special:*** *A former French colony with a unique blend of Indian and French cultures.*
- ***Attractions:*** *Promenade Beach, Sri Aurobindo Ashram, Auroville, Paradise Beach, French Quarter.*
- ***Budget Considerations:*** *Pondicherry has a range of accommodation options, from budget guesthouses to more upscale hotels. Food is relatively inexpensive.*
- ***Best Season:*** *October to March. February is a great time to visit.*
- ***Ideal Days:*** *3-4 days.*
- ***Travel from Kolkata:*** *Take a train to Chennai, then a bus or taxi to Pondicherry (approx. 3-4 hours). Total train travel around 24-26 Hours.*

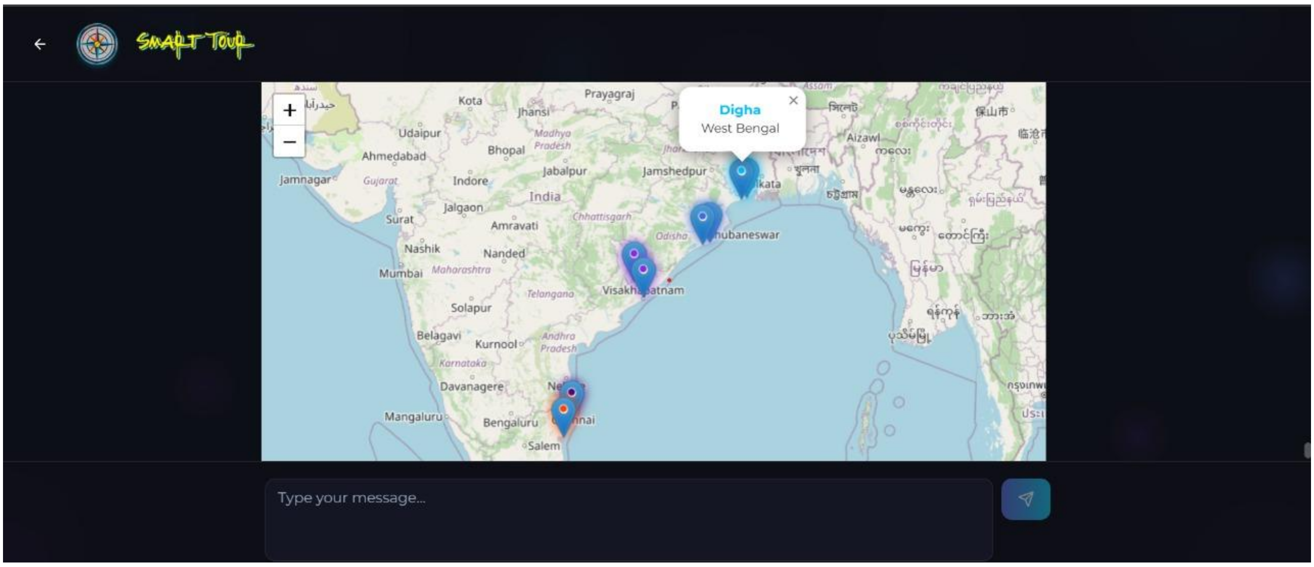
***Budget Breakdown (Approximate for 7 Days):***

- ***Transportation: 2500 INR***
- ***Accommodation: 2500 INR***
- ***Food: 2000 INR***
- ***Sightseeing & Activities: 1500 INR***
- ***Miscellaneous: 1500 INR***

***Travel Tips for February:***

- *Book transportation and accommodation in advance, especially if traveling during peak season.*
- *Pack light cotton clothing, sunscreen, sunglasses, and a hat.*
- *Stay hydrated by drinking plenty of water.*
- *Bargain while shopping in local markets.*
- *Be mindful of local customs and traditions.*
- *Carry a valid ID proof.*
- *Check weather forecasts before traveling.”*

## *Destinations Mapped:*



# CONCLUSION

The Smart Tour expert system demonstrates how AI can meaningfully enhance user experience in real-world domains like travel. By replacing rigid logic rules with language-based AI, it delivers contextual, conversational, and adaptive trip planning. The project shows how combining Gemini LLM for recommendation, regex-based NLP for entity extraction, and OpenCage geocoding with Leaflet visualization can produce a fluid, intelligent decision-making pipeline. It highlights the shift from symbolic AI (e.g., Prolog) to data-driven, probabilistic systems capable of deeper user understanding. Overall, this work showcases the practical application of expert systems in everyday use, particularly for personalized decision support.